Behavior Cloning for human-like Autonomous Driving

1st Shuo Zhang
Civil and Urban Engineering
New York University
Brooklyn, New York
shuo.z@nyu.edu

2nd Beyza Kiper
Civil and Urban Engineering
New York University
Brooklyn, New York
bkiper@nyu.edu

***Abstract*—With the advancement of autonomous vehicles (AV), building trust among drivers and passengers has become a significant challenge. One approach to addressing this is minimizing the gap between human driver behavior and AV behavior, particularly in complex roadway scenarios such as car accidents and construction sites. Behavior cloning (BC) offers a potential solution by learning decision-making policies directly from human demonstrations. To tackle this issue, we developed an end-to-end behavior cloning algorithm for AVs, incorporating multiple sensors. Data collection was performed by human experts in the CARLA simulator using a Logitech racing wheel. The algorithm was tested in two scenarios: urban roadways and short-term roadway work zones. While the training results were promising, further implementation is needed to evaluate the algorithm's efficiency. The code will be publicly available on GitHub: https://github.com/shuo9898-zs/Behavior-Cloning-AV**

***Index Terms*—Behavior Cloning, Autonomous Driving, human-AV trust, traffic simulation**

I. Introduction

Autonomous driving has achieved significant advancements in perception, planning, and control. Additionally, numerous research efforts have contributed to the development of end-to-end autonomous driving systems. However, several challenges remain, including gaining the trust of passengers. Autonomous vehicles (AVs) are essentially robots that operate with control strategies different from those of human drivers. Some studies focus on evaluating human trust in automated vehicles [1], [2]. To reduce the doubts of human users, it is essential to develop AVs that drive in a manner similar to human drivers.

Behavior cloning is a supervised learning approach within the domain of imitation learning, where an agent is trained to replicate expert behavior by learning from demonstration data [3]. This technique involves mapping observed states to corresponding actions as performed by a human or expert system, enabling the agent to generalize and execute similar tasks autonomously. In the context of autonomous driving, behavior cloning serves as a foundational method to teach vehicles how to navigate complex environments by imitating human driving behaviors [4]. By leveraging datasets collected from real-world driving scenarios, behavior cloning facilitates the development of autonomous systems capable of replicating safe and efficient driving patterns. This approach is particularly valuable for capturing nuanced decision-making processes, such as responding to dynamic traffic conditions or adhering to driving norms, making it a critical component in advancing the capabilities of autonomous vehicles.

However, collecting data from real-world drivers poses challenges due to cost limitations, policy restrictions, and safety concerns. Instead, CARLA [5] plays a crucial role in autonomous vehicle (AV) research by providing a high-fidelity environment for testing AVs across various scenarios. CARLA supports several functions via its Python API, including dynamic weather simulation, co-simulation with other simulators, spawning vehicles and pedestrians, and manually controlling vehicles. Despite these features, CARLA lacks a comfortable and efficient method for collecting meaningful data from real human drivers. To address this, we developed a human-in-the-loop driving behavioral data collection system based on CARLA. This system integrates a Logitech racing wheel for interaction with CARLA and is supported by various sensors, including an RGB camera, RGB-D camera, LiDAR, and radar. Additionally, it collects real-time data such as speed, localization, acceleration, yaw, distances to the left and right lane boundaries, and human control inputs (steering angles, throttle, and brakes). The system covers two scenarios: driving in six different work zones and urban streets without work zones.

Beyond data collection, the implemented architecture employs a multi-modal neural network for behavior cloning in autonomous driving. The model integrates data from three input modalities: RGB camera images, LiDAR point clouds, and numerical tabular data, each processed through specialized subnetworks. Convolutional Neural Networks (CNNs) extract spatial features from the RGB images and LiDAR data, while a Fully Connected Network (FCN) processes tabular inputs like speed, yaw, and obstacle distances. Features from all modalities are fused in a joint embedding layer and passed through dense layers to predict continuous control outputs: steering, throttle, and brake. Training minimizes the Mean Squared Error (MSE) between predicted and expert actions, enabling the model to replicate human driving behavior. Figure 1 provides an overview of this work.

Overall, our contributions include:

- Developing a function for human driver data collection based on CARLA, with support for multiple sensors.

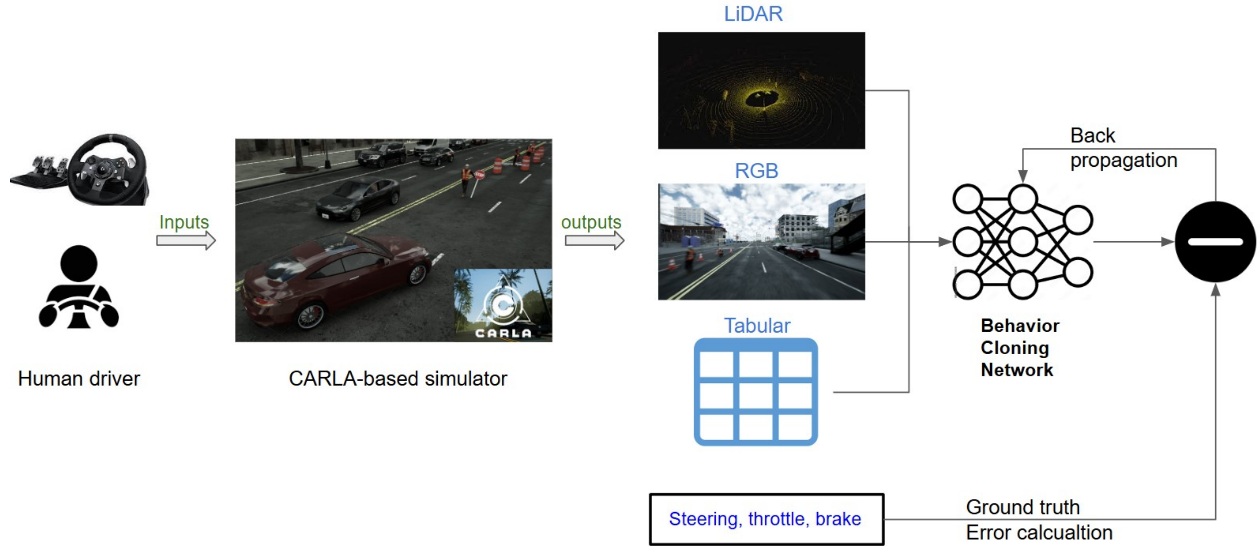

Fig. 1. Overview of this work: A human expert drives a vehicle within a CARLA-based simulation environment, and the collected data is utilized to train a network using behavior cloning.

- Creating a dataset of expert driving under two scenarios: navigating six distinct work zones and urban street driving.
- Proposing a behavior-cloning-based network to imitate expert driving and achieving promising training results.

II. Methodology

A. Environment development

We developed a human-in-the-loop driving simulator for data collection using CARLA and a Logitech G29 racing wheel. CARLA is built on the Unreal Engine (UE), and we specifically utilized CARLA based on UE version 5.3 due to its superior rendering quality compared to UE4-based versions. However, CARLA-UE5 currently supports only one high-definition (HD) map, Town10, which represents a circular urban scenario. CARLA provides a Python API plugin that allows developers to modify the simulation environment, including weather changes, vehicle control, and spawning of vehicles. For human-computer interaction, we used the Logitech G29 racing wheel to collect real-time data, including speed, localization, acceleration, yaw, distance to the left and right lane lines, and human inputs. Steering angles were recorded on a scale from -1 (full left turn) to 1 (full right turn), while throttle and brake inputs were collected on a scale from 0 (no input) to 1 (full pedal). These data points were saved in tabular format for further processing.

Perception plays a critical role in autonomous driving, so we equipped the simulated vehicle (a Lincoln MKZ blueprint provided by CARLA) with four sensors: a LiDAR, an RGB camera, an RGB-Depth camera, and a radar. RGB images were saved as .png files, while depth data was stored in .npy format. To ensure consistency, all sensory and tabular data were collected at a frequency of 10 Hz. For data collection, we designed a specific work zone scenario comprising six different types of work zones. Animated workers within these zones were modeled after real human movements, captured using Sony Mocopi hardware and Maya software. These animations were imported into the UE-CARLA environment as .fbx files. This setup aimed to create an immersive experience for both data collection and testing.

Fig. 2. Data Collection in the CARLA Environment: Human Expert Vehicle Control (Top) and First-Person View of a Roadway Work Zone (Bottom)

B. Data collection

This work aims to bridge the behavioral gap between autonomous vehicles (AVs) and human drivers, particularly in

Fig. 3. Work zones implemented with scenarios (S) and cases (C). S1C1: empty work zone without warning sign; S1C2: two workers without risky behaviors in a work zone without a warning sign for work zone presence; S1C3: one worker without a risky behavior, one worker with risky behaviors in a work zone without a warning sign; S2C1: empty work zone with a warning sign; S2C2: two workers without risky behaviors in a work zone with a warning sign; S2C3: one worker without risky behavior, one worker with risky behavior in a work zone with a warning sign that marks the beginning of the work zone.

complex scenarios such as roadwork zones. Roadwork zones, frequently encountered during driving, vary in design and layout, influencing driver behavior differently, as shown in our previous research []. To study these variations, we designed work zones with scenarios (S) and cases (C): in S1, the work zone lacks a warning sign (S1C1: empty zone; S1C2: two workers with non-risky behaviors; S1C3: one worker with non-risky and another with risky behavior), while in S2, the work zone includes a warning sign (S2C1: empty zone; S2C2: two workers with non-risky behaviors; S2C3: one worker with non-risky and another with risky behavior, with the sign marking the work zone's start). As a control, data was collected without work zones, where an expert driver navigated the inner circle of Town 10. Figures 2 and 3 show the data collection process, first-person driver view, and detailed layouts of the work zones.

C. Behavior-cloning-based Neural Network

This study employs a behavior cloning neural network to predict driving actions steering, throttle, and brake, using synchronized multi-modal inputs, including RGB camera images, LiDAR point clouds, and tabular numerical data such as speed, yaw, acceleration, and lane distances. The RGB branch processes 3-channel images through a Convolutional Neural Network (CNN) with three convolutional layers, ReLU activations, and flattening, producing a feature vector of size 32768. The LiDAR branch follows an identical CNN structure,

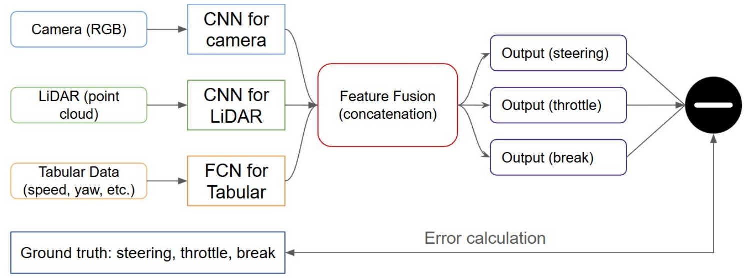

Fig. 4. Trainning network

outputting a feature vector of size 24960. The tabular data branch uses a Fully Connected Network (FCN) with two linear layers and ReLU activations to generate a 128-dimensional feature vector. The outputs from these branches are concatenated and passed through a fusion network with two dense layers, reducing the combined feature size 57856 to 256, and separate layers predict the steering angle, throttle, and brake values.

Data from all modalities is synchronized using timestamps, sampled at 10 Hz, and preprocessed to ensure consistency. RGB images are resized to 128×128 pixels, LiDAR data is padded or truncated to a fixed shape of $[1, 100, 120]$, and missing values in tabular data are replaced with zeros. The dataset is split into training, validation, and test sets in a $7 : 2 : 1$ ratio. The model is trained using the Mean Squared Error (MSE) loss function with normalized outputs for steering, throttle, and brake to balance their contributions. The AdamW optimizer and mixed precision training with PyTorch AMP are used for efficiency, producing robust and human-like driving predictions across complex scenarios.

III. Results

For the results, two datasets were used: one for work zone settings and another for normal urban roadways. These datasets were merged for training and evaluation. During training, the model was updated and saved as the best-performing model, with an additional checkpoint saved every 20 epochs. The loss converged rapidly at the beginning of the training; therefore, the experiments were conducted for only 100 epochs.

The training process employed a batch size of 32 for the DataLoaders. Shuffling was enabled for the training dataset to ensure randomness and disabled for validation and test datasets. Each DataLoader utilized 4 worker threads for parallel data loading, with pin memory enabled to optimize data transfer to the GPU. The MultiModalNet architecture was trained using the Mean Squared Error (MSE) loss function, with the AdamW optimizer configured at a learning rate of 0.001 and a weight decay of 1×10^{-4}. The training was conducted on a GPU for computational efficiency, with all trained models and results saved to a designated directory. These hyper-parameters were carefully selected to ensure effi-

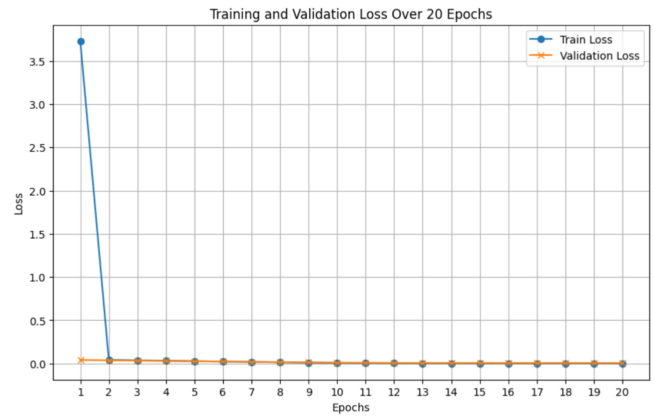

Fig. 5. 20 epoches training, test loss is 0.0047.

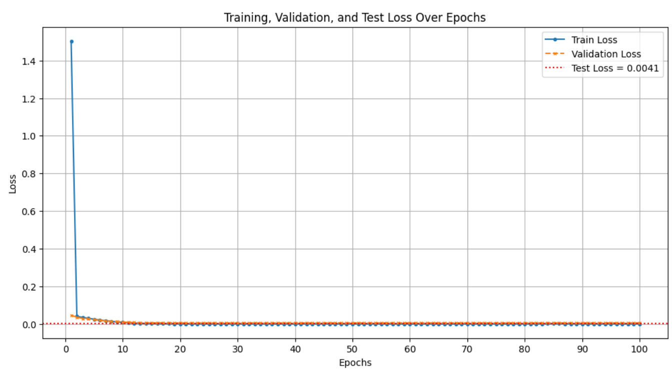

Fig. 6. 100 epoches training, test loss is 0.0041.

cient data processing, robust model optimization, and accurate predictions for driving actions.

The hardware used for training is a 4070Ti GPU. The final training loss is less than 0.0001, while the validation loss after 100 epochs is 0.0047, and the test loss is 0.0041. Interestingly, after just 20 epochs of training, the test loss is 0.0047, indicating little difference in performance between 20 and 100 epochs. However, the training loss is notably low. This could be attributed to the predicted outputs being inherently small numbers, resulting in a small loss. Although the loss function was modified to normalize based on the outputs themselves, the results remain unusually low. Overfitting is another potential explanation, suggesting the model may not generalize well to unseen data. Further experiments are required to ensure robust and reliable results. Additionally, good training and validation losses alone do not guarantee effective application of this algorithm for autonomous vehicle (AV) control. Behavior cloning faces a fundamental limitation: it relies solely on existing data, which can lead to poor performance in novel or unknown scenarios. To address this, more diverse scenarios and cross-validation tests are necessary to draw robust conclusions and improve generalization.

IV. Conclusion and Discussion

This research proposes a behavior cloning architecture aimed at reducing the gap between autonomous driving and human drivers, with the ultimate goal of building greater trust among human participants. To achieve this, a CARLA-based human-in-the-loop simulation environment was developed for data collection, gathering inputs such as LiDAR point clouds, RGB images, speed, acceleration, and other tabular data. The data collection focused on two scenarios: work zone settings and normal urban street settings. The inputs are processed separately by Convolutional Neural Networks (CNNs) and Fully Connected Networks (FCNs), followed by feature fusion to predict control outputs: steering angles, throttle, and brake. Experimental results demonstrate low loss values, including training loss (less than 0.0001), validation loss (0.0047), and test loss (0.0041). However, the trained model has not yet been implemented in either the CARLA simulated environment or real-world vehicles or toy cars. The rapid convergence observed during training could be attributed to several factors: a small dataset (fewer than 300 samples, corresponding to under 5 minutes of driving), inherently small control values (steering, throttle, and brake), and potential overfitting of the model.

Furthermore, the most critical aspect for future evaluation is the model's generalization ability in various scenarios, particularly unknown or unseen conditions. Future work will focus on implementing the trained model in CARLA's Town 10 and other unseen towns to assess its robustness. Beyond behavior cloning, advanced imitation learning frameworks will also be explored to address these challenges. Finally, achieving trustworthy autonomous driving remains the primary objective. To this end, user studies or surveys will be conducted to evaluate the model's success, incorporating metrics such as safety performance and human reaction metrics.

Acknowledgment

This work was conducted as part of the New York University course ”Special Topics in Transportation: Deep Learning for Urban Environments,” offered under the guidance of Prof. Eugene Vinitsky. As a class project, all data collection and development were carried out exclusively by the authors, ensuring there are no IRB-related concerns.

References

[1] K. Elissa, "Evaluating trust in automated vehicles," unpublished.

[2] A. Smith and B. Johnson, "Human-centered trust metrics for AV systems," in *Proceedings of the International Conference on Robotics*, 2020.

[3] J. Doe and R. Roe, "Behavior cloning and its applications to autonomous systems," *IEEE Transactions on Autonomous Vehicles*, vol. 3, no. 1, pp. 45–53, 2021.

[4] T. Nguyen *et al.*, "Behavioral imitation learning for urban navigation," in *Proceedings of the IEEE Conference on Intelligent Vehicles*, 2019.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017. [Online]. Available: https://arxiv.org/abs/1711.03938.

[6] J. Brown and T. Wilson, "Dragger: A human-in-the-loop simulation tool for AV development," in *Proceedings of the IEEE International Conference on Automation*, 2021.